50325-0633 (Seq. No. 5016) *Patent*

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR COLLECTING, AGGREGATING AND MONITORING NETWORK MANAGEMENT INFORMATION

INVENTORS:

RITURAJ KIRTI
JOHN PARELLO

PREPARED BY:

HICKMAN PALERMO TRUONG & BECKER LLP
1600 WILLOW STREET
SAN JOSE, CA 95125
(408) 414-1080

EXPRESS MAIL MAILING INFORMATION

"Express Mail" mailing label number EL734779978US

Date of Deposit February 13, 2002

METHOD AND APPARATUS FOR COLLECTING, AGGREGATING AND MONITORING NETWORK MANAGEMENT INFORMATION

FIELD OF THE INVENTION

[0001] The present invention generally relates to data processing. The invention relates more specifically to computer systems or software systems that collect, aggregate and monitor network management information.

BACKGROUND OF THE INVENTION

[0002] A computer network generally includes a number of network devices, such as switches, routers, and others, connected so as to allow communication among the devices and end station devices such as desktop machines, servers, hosts, printers, fax machines, and others. Each network device has a processor and a memory. Status variables and other values in the memory are continuously changed and updated as the device operates.

[0003] Computer networks have become ubiquitous in the home, office and industrial environment. As computer networks have grown ever more complex, automated mechanisms for organizing and managing the networks have emerged. These mechanisms are generally implemented in the form of one or more computer programs, and are generically known as network management systems or applications. Network management refers to the function of requesting and monitoring information on a network device relating to fault, configuration, status, performance and security. To monitor the status of a device in the network, a network management station or system transmits a message requesting information over the network to a software program or agent running on the target network

device. In response, the agent sends a message over the network to the network management station. The communications are carried out according to an agreed-upon protocol, such as the Simple Network Management Protocol (SNMP).

[0004] However, the hardware and software that perform network management functions must often deal with many different network devices that communicate using device-specific protocols. Thus, the scalability of a network management system can be limited when new devices are added to the network with protocols that are not supported by the system. Also, network management systems can only return information about the current status of a device at the time the information is requested, in essence forming a "snapshot" of the status of the device. Finally, current network management systems lack the capability to analyze and monitor the information that is collected from network devices.

[0005] One past approach is to use network management data models and information bases in an attempt to normalize the desired information. Examples of such models include the Cisco Information Model, as implemented in the Asynchronous Network Interface element of certain network management systems from Cisco Systems, Inc. as well as other proprietary systems. These systems are capable of identifying devices in a network and gathering data from a device.

[0006] However, these approaches have numerous disadvantages. First, they are difficult to scale when changes occur, usually requiring a software upgrade, and cannot model all the information desired to do network management. Occasionally, committee reviews are required to approve the changes. Second, a network device may require separate network management software in order to accommodate the many different network communication protocols used to extract information from the device when it is not present in the network management models or databases. Third, models can only show the state of the network

device at the moment the status is requested, and cannot generate information as to the performance of a device over time. This "snapshot" information is less meaningful to a user than is data that shows a trend between two or more data points in time. Finally, these models cannot generate a notification to a user when data falls outside a desired value or range of values.

[0007] Based on the foregoing, there is a clear need for a method of network management that can specify what device data to collect, where to collect it, when to collect it and how to analyze it after collection, without the need to upgrade the network management software. Also, there is a need for a method that is scalable with various network devices using different network communication protocols and can monitor network device activity.

[0008] The approaches described in this section could be pursued, but are not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated herein, the approaches described in this section are not prior art to the claims in this application and are not admitted to be prior art by inclusion in this section.

SUMMARY OF THE INVENTION

[0009] The foregoing needs, and other needs and objects that will become apparent for the following description, are achieved in the present invention, which comprises, in one aspect, a method for collecting, aggregating and monitoring network management information in a network management system. The method utilizes user-definable configuration information comprising an operational specification on what to collect, aggregate and monitor. The method can change the operational specification without software upgrades. Included in the operational specification are scheduling instructions for carrying out one or more of these tasks at preset times or intervals.

[0010] The method begins by identifying network devices on the network. The network devices are then queried and data is acquired in accordance with the instructions contained in the operational specification of the configuration information. The method can perform transformations to the acquired data according to formulas specified in the operational specification. The method periodically monitors the data for compliance with specific threshold conditions, which are also a part of the operational specification. It can then generate a notification to a user whenever a threshold condition has been met. The acquired or transformed data can be stored to a database for subsequent retrieval. The operational specification contains instructions on how to aggregate the stored data in order to generate trending information. Data stored in the database can be subsequently removed automatically based on aging criteria or other criteria specified by the user. The method can render data to a display for viewing by the user using device-specific GUI attributes.

[0011] In other aspects, the invention encompasses an apparatus and a computer readable medium configured to carry out the foregoing steps.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0013] FIG. 1 is a block diagram of a network that is managed by a network management system running on one or more network management applications;

[0014] FIG. 2 is a block diagram that illustrates an overview of a network management collector;

[0015] FIG. 3A is a block diagram that illustrates the structure of an XML configuration file;

[0016] FIGS. 3B and 3C depict examples of an XML DTD and XML file, respectively;

[0017] FIG. 4 is a block diagram that illustrates an overview of a collector module;

[0018] FIG. 5 is a block diagram that illustrates an overview of an aggregator module;

[0019] FIG. 6 is a block diagram that illustrates an overview of a reaper module;

[0020] FIG. 7 is a block diagram that illustrates an overview of the display module;

[0021] FIGS. 8A and 8B are flow diagrams describing a method for collecting, aggregating and monitoring network management information; and

[0022] FIG. 9 is a block diagram that illustrates a computer system upon which an aspect of the invention may be implemented.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0023] A method and apparatus for collecting, aggregating and monitoring network management information is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0024] Embodiments are described herein according to the following outline:

1.0 OPERATIONAL CONTEXT

2.0 STRUCTURAL OVERVIEW

3.0 FUNCTIONAL OVERVIEW

4.0 IMPLEMENTATION MECHANISMS -- HARDWARE OVERVIEW

5.0 EXTENSIONS AND ALTERNATIVES

1.0 OPERATIONAL CONTEXT

[0025] FIG. 1 is a simplified diagram of a network 100 that is managed by a network management system running on one or more network management stations 10. The network 100 comprises one or more network devices 102, such as switches, routers, bridges, gateways and other devices. Each network device 102 is coupled to another network device 102, or to one or more end stations 120. Each end station 120 is a terminal node of the network 100 at which some type of work is carried out. For example, an end station 120 is a workstation, a printer, a server, or similar device.

[0026] Each network device 102 executes a network-oriented operating system 110. An example of a network-oriented operating system is the Internetworking Operating System (IOS) commercially available from Cisco Systems, Inc. Each network device 102 also executes one or more applications 112 under control of the operating system 110. The operating system 110 supervises operation of the applications 112 and communicates over network connections 104 using one or more agreed-upon network communication protocols, such as Simple Network Management Protocol (SNMP).

[0027] Each device 102 stores information about its current configuration, and other information, in one or more forms of organized data storage, for example, a Management Information Base (MIB) 114. Information in the MIB 114 is organized in one or more MIB variables. The network management station 10 can send "get" and "set" commands to the device 102 in order to retrieve or set values of MIB variables. Examples of MIB variables include *sysObjectID* and *sysDescr*. For information stored in other forms, there are other types of communications and commands to set and retrieve the information values.

[0028] The network management station 10 may be a general-purpose computer system of the type shown and described further herein in connection with FIG. 3. The network management station 10 executes one or more software components that carry out the functions shown in block diagram form in FIG. 1. For example, the network management station 10 executes a basic input/output system (BIOS) 20 that controls and governs interaction of upper logical layers of the software components with hardware of the network management station. An example of a suitable BIOS is the Phoenix ROM BIOS. The network management station 10 also executes an operating system 30 that supervises and controls operation of upper-level application programs. An example of a suitable operating system is the Microsoft Windows NT® operating system. The network management station

10 may also execute other operating systems that may not require a BIOS 20, such as UNIX-type operating systems, microkernel-based operating systems, etc.

[0029] The network management station 10 executes an asynchronous network interface (ANI) 50 under control of the operating system 30. The ANI 50 provides an interface to the network 100 and communicates with the network using SNMP or other agreed-upon protocols. The ANI 50 provides numerous low-level services and functions for use by higher-level applications.

[0030] The network management station 10 executes a network management system 40 that interacts with an information base 60 containing information about the managed network 100. The information base may be implemented on one or more of: relational databases, object databases, directories, flat file systems, ISAM file systems, etc. The network management system 40 is an example of a network management application. Using a network management application, a manager can monitor and control network components. For example, a network management application enables a manager to interrogate devices such as host computers, routers, switches and bridges to determine their status and to obtain statistics about the networks to which they attach. The network management application also enables a manager to control such devices by changing device configuration or operation information, for example, routes and configuring network interfaces. An example of a network management application is Resource Manager Essentials, commercially available from Cisco Systems, Inc.

[0031] The ANI 50 and network management system 40 need not execute or reside on the same physical computer. They may execute on different machines. There need not be only one ANI 50 or only one network management system 40.

2.0 STRUCTURAL OVERVIEW

[0032] FIG. 2 is a block diagram that illustrates an overview of a network management collector 200. The network management collector 200 may be implemented as a part of the network management system 40 (FIG. 1). As can be seen in FIG. 2, the network management collector 200 includes configuration information 210, at least one collector module 220 (each including a monitor module 225), at least one aggregator module 230, at least one reaper module 240 and a display module 250. As a component of the network management system 40, the network management collector 200 interfaces with the network devices 102 and the information base 60. In one aspect of the invention, the information base 60 further includes a short-term database 260, a long-term database 270 and a device inventory database 280.

[0033] In one aspect of the invention, configuration information 210 is an XML (Extensible Markup Language) file that contains data on how to configure and operate the collector module 220, the aggregator module 230, the reaper module 240 and the display module 250. The function of configuration information 210 is described in more detail below with respect to FIG. 3.

[0034] The collector module 220 is responsible for gathering data from network devices 102 for a particular network management purpose. For example, the data may include, but is not limited to, information relating to status or performance of each network device 102. The collector module 220 can perform periodic queries on the data, transform the data according to preset formulas and store resultant data to the short-term database 260. The collector module 220 can also monitor the data for compliance with prescribed threshold conditions. All data acquisition scheduling, queries, transforms, storage instructions and monitoring

thresholds are specified in the configuration information 210. The function of the collector module 220 is described in more detail below with respect to FIG. 4.

[0035] The aggregator module 230 is responsible for periodically selecting and transforming data from one of the collector modules 220 and creating trending information. The aggregator module 230 stores resultant trending information to the long-term database 270. Data transforms, trending formula and storage instructions are specified in configuration information 210. The function of the aggregator module 230 is described in more detail below with respect to FIG. 5.

[0036] The reaper module 240 is responsible for removing data from the short-term database 260 and the long-term database 270 when it is no longer desired. Data is removed based on parameters specified in configuration information 210. The function of the reaper module 240 is described in more detail below with respect to FIG. 6.

[0037] The display module 250 is responsible for rendering a user visible data display of information stored in the short-term database 260 and the long-term database 270. The information to be displayed and the format of the display are based on parameters specified in configuration information 210. The function of the display module 250 is described in more detail below with respect to FIG. 7.

3.0 FUNCTIONAL OVERVIEW

[0038] FIG. 3A depicts a block diagram that illustrates the structure of configuration information 210 when the configuration information is structured as an XML file. XML files provide a flexible way to create common information formats and share both the format and data on various networks, such as the World Wide Web and intranets, and can be processed purely as data by a program, stored with similar data on another computer or displayed.

Using an XML file as configuration information 210 allows a user to determine what, where, when and how data is to be collected in a fast and flexible manner without the need to upgrade the network management software.

[0039] As shown in FIG. 3A, configuration information 210 is divided into various sections that prescribe the configuration and operation of one or more modules of the network management collector 200. In one aspect of the invention, the sections include, but are not limited to, a name section 310, a scheduler section 320, a target section 330, an operation section 340, a storage section 350, a threshold section 360, an aggregation section 370, a table trimmer section 380 and a presentation section 390. Each configuration information 210 thus forms a unique data acquisition set, configured for performing specific measurements, operations and tasks on specific network devices.

[0040] Information stored in name section 310 of configuration information 210 assigns a distinctive name to the file, thereby identifying it as a unique data acquisition set. The collector module 220 can thereby distinguish the various files, or data acquisition sets, by name to assure that the desired data is acquired at the desired time from the desired device or devices.

[0041] The scheduler section 320 of configuration information 210 contains instructions on when and how often the collector module 220 is to acquire data. Polling of network devices 102 can thus be scheduled at prescribed time intervals, at specific dates and times, or at specific recurring frequencies.

[0042] The target section 330 of configuration information 210 contains two basic types of information. First, it contains information as to the identity of the specific network devices 102 that the collector module 220 will poll. Thus, a user can pick and choose which network devices 102 to poll and which to exclude from polling. Second, target section 330

contains device-specific information related to data type and protocol for each network device 102. This information includes what data types are available for collection from each device (such as CPU usage, device cache hits, dropped packets, etc.), which network communication protocol is required to collect the data type (such as SNMP, Telnet, etc.), and device-specific graphical user interface (GUI) attributes.

[0043] The operation section 340 of configuration information 210 contains one or more arithmetic formulas that define specific transformations to be performed on the acquired data. Storage section 350 contains instructions relating to storage of values resulting from the transformations, such as specifying the row and column of the database where the data should be stored.

[0044] The threshold section 360 of configuration information 210 contains threshold criteria values. Such criteria may include, but is not limited to, specified minimums, maximums, averages, etc. Additionally, threshold section 360 contains information defining if and when to generate a notification 440 upon the occurrence of a threshold condition, discussed below with respect to FIG. 4.

[0045] The aggregation section 370 of configuration information 210 contains various parameters relating to aggregating the acquired data, such as how often stored data should be aggregated, which data to select, specific formulas for transforming the data into trending information and how to store the resultant information to the long-term database 270.

[0046] The table trimmer section 380 of configuration information 210 contains instructions for removing or trimming data from either the short-term database 260 or the long-term database 270.

[0047] The presentation section 390 of configuration information 210 contains an application program interface (API) that provides presentation logic to the system. The

presentation logic performs various tasks, such as identifying the rows and columns of data that was collected, determining where the data is located, obtaining the GUI attributes of the network device 102 from which the data was taken, specifying which rendering algorithm to use, etc. The API thus uses the presentation logic to render a GUI so that data can be displayed to a user.

[0048] In one aspect of the invention, an XML document type definition (DTD) file is used to set up the configuration information 210. The XML DTD is the specification that identifies the markups in the configuration information 210 for various functions, such as separating paragraphs, identifying topic headings, etc., and determining how each is to be processed. FIG. 3B is a table showing a listing of an exemplary XML DTD file. FIG. 3C is a table showing a listing of configuration information 210 as an exemplary XML configuration file using the XML DTD file of FIG. 3B.

[0049] As shown in FIG. 3C, a name segment 311 of configuration information 210 contains information on the identity of the unique data acquisition set corresponding to name section 310. In this example, the name of the XML configuration file is "Router Host Information". A scheduler segment 321 contains the schedule for the polling frequency corresponding to scheduler section 320 of configuration information 210. In this example, the scheduler segment contains instructions to poll the network devices every 60 seconds. A segment 331 contains the identity of the particular network devices to be polled corresponding to target section 330 of configuration information 210. In this example, the target devices are routers. A segment 332 performs acquisition of specific data on the target devices. A segment 341 contains instructions on formulas and transformations that are to be performed on the data corresponding to operation section 340 of configuration information 210. A segment 351 contains instructions on outputting and storing the acquired or

transformed data corresponding to storage section 350 of configuration information 210. A segment 361 sets threshold limits and segment 362 contains information on generating a notification corresponding to threshold section 360 of configuration information 210.

[0050] FIG. 4 is a block diagram that illustrates an overview of the operation of the collector module 220 of the network management collector 200. The collector module 220 retrieves an inventory of available network devices 102 from the device inventory database 280. Because the collector module 220 is running as a part of the network management system 40 of network management station 10 (FIG. 1) as a node on the network, it can communicate with each available network device 102. Before acquiring data from the available network devices 102, however, the collector module 220 must parse the pertinent sections of configuration information 210 for configuration and operational instructions.

[0051] The collector module 220 parses information from scheduler section 320 of configuration information 210, then executes the instructions by way of a scheduler block 410. The function of scheduler block 410 is to tell the collector module 220 when to poll the network devices 102 for specific data. As discussed above, polling can be scheduled at prescribed time intervals, at specific dates and times, or at specific recurring frequencies. For example, the scheduler block 410 can instruct the collector module 220 to poll devices every six minutes, on the fourth Thursday of each month, etc.

[0052] The collector module 220 further includes a query block 420. The query block 420 causes the collector module 220 to poll and query specific targeted network devices 102 specified by the information parsed from target section 330. Each network device 102 is queried using the protocol-specific variables for that device at the times specified by scheduler block 410. For example, target section 330 can instruct query block 420 to query only routers on the network and specify the appropriate SNMP variable "ipVariable".

[0053] After each network device 102 is polled and queried, the data thus acquired is sent to an output block 430 of the collector module 220. The output block 430 serves two purposes. First, it executes one or more simple or complex arithmetic operations on the acquired data, producing a data transformation. The specific transformation is defined by one or more formulas that are parsed from operation section 340 of configuration information 210. The formulas can be applied to any or all data acquired by the query block 420. Second, output block 430 will store the acquired data, the transformed data or both to the short-term database 260. Instructions for data storage originates from information parsed from storage section 350 of configuration information 210.

[0054] The monitor module 225 of the collector module 220 verifies whether the acquired data, the transformed data or both meet specific threshold criteria. The threshold criteria are defined by information parsed from threshold section 360 of configuration information 210. As discussed above, such criteria may include specified minimums, maximums, averages, etc. Additionally, threshold section 360 contains information defining if and when to generate notification 440 upon the occurrence of a threshold condition. For example, a user can generate notification 440 when data falls outside of desired boundaries. The notification 440 can be stored in the short-term database 260 or the long-term database 270.

[0055] FIG. 5 is a block diagram that illustrates an overview of the aggregator module 230 of the network management collector 200. The function of the aggregator module 230 is to aggregate stored data into information that shows data trends over time. This trending information can give a user a long-term view of the stored data and reveal any patterns that may exist.

[0056] To provide the desired aggregation of stored data, the aggregator module 230 parses instructions from scheduler section 320 and aggregation section 370 of configuration information 210. As discussed above, scheduler section 320 and aggregation section 370 contain various parameters on aggregating the acquired data, such as how often stored data should be aggregated, which data to select, specific formulas for transforming the data into trending information and how to store the resultant information to the long-term database 270. In accordance with the parsed instructions, the aggregator module 230 periodically reads data from the short-term database 260 stored by the collector module 220 (FIG. 4) as described above, transforms the stored data into trending information, and stores the resultant trending information to the long-term database 270. For example, the collector module 220 can store data averages every six minutes into short-term database 260, and the aggregator module 230 can save the total data average at the end of the day into long-term database 270.

[0057] FIG. 6 is a block diagram that illustrates an overview of the reaper module 240. The purpose of the reaper module 240 is to remove or trim data stored on the short-term database 260 and the long-term database 270 once such data is no longer needed. As discussed above, the instructions and the criteria for trimming data from either database is parsed from information contained in table trimmer section 380 of configuration information 210. For example, table trimmer section 380 may instruct reaper module 240 to delete any data older than 30 days from short-term database 260, and data more than 90 days old from long-term database 270.

[0058] FIG. 7 is a block diagram that illustrates an overview of the display module 250. The display module 250 causes information to be presented to a user by way of a display device 710. An example of display device 710 is display 812, shown below in reference to FIG. 8. The display module 250 parses information contained in target section 330 and

presentation section 390 of configuration information 210. As discussed above in reference to FIG. 3, presentation section 390 contains an application program interface (API) that provides presentation logic to the system. The presentation logic performs various tasks, such as identifying the rows and columns of data that was collected, determining where the data is located, obtaining the GUI attributes of the network device 102 from which the data was taken, specifying which rendering algorithm to use, etc. The API thus uses the presentation logic to render a GUI so that data can be displayed to a user.

[0059] FIGS. 8A and 8B are flow diagrams describing a method for collecting, aggregating and monitoring network management information. The process starts with step 805 of FIG. 8A, where configuration information 210 is configured. The configuration information 210 is preferably an XML file. After all XML configuration files are configured, in step 810 a specific XML configuration file is selected for execution. The name, scheduler, target, operation, storage and threshold sections of the XML configuration file are parsed by the collector module 220 in step 815, in the manner described above in reference to FIG. 4.

[0060] In step 820, the process inquires if a scheduled acquisition of data is due. The scheduling of an acquisition is parsed from scheduler section 320. If no acquisition is due, then at step 821 the process waits a predetermined period of time and inquires again at step 820. If an acquisition is due, then in step 825 the process identifies specific target network devices 102 for data acquisition from information parsed from target section 330. In step 830, the process causes data to be acquired from target network devices 120.

[0061] In step 835, the process inquires whether the just acquired data should be transformed according to instructions parsed from the operation section 340. If transformation is required, then in step 836 the process causes the data to be transformed accordingly. Then, in step 840, the resultant data is stored in the short-term database 260

according to instructions parsed from storage section 350. If no transformation is required, the process bypasses step 836 and in step 840 simply stores the just acquired data in the short-term database 260.

[0062] In step 845, the process inquires whether the acquired data and/or transformed data meet specific threshold criteria based on information parsed from threshold section 360. If the threshold criteria are met, then in step 846 the process generates a notification 440 (FIG. 4). In step 847, the notification 440 is stored in the short-term database 260 or the long-term database 270 according to instructions from storage section 350. The process continues at step 850, described below. If the threshold criteria are not met, then the process bypasses steps 846 and 847 and proceeds to step 850.

[0063] As shown in FIG. 8B, step 850 parses instructions from aggregation section 370 as described above in reference to FIG. 5. In step 855, the process uses this information to inquire whether the just acquired data should be aggregated. If so, then in step 856 the process causes data to be read from the short-term database 260 and aggregates the data into trending information according to the instructions parsed from aggregation section 370. In step 857, the resultant trending data is then stored in the long-term database 270. If the data is not to be aggregated, then the process bypasses steps 856 and 857 and continues in step 860.

[0064] In step 860, the process parses instructions from table trimmer section 380 as described above in reference to FIG. 6. In step 865, the process uses this information to inquire whether any of the data currently stored in either the short-term or the long-term database should be removed. If data is identified for removal, then in step 866 the process causes the data to be deleted from either the short-term database 260, the long-term database

270, or both, according to the instructions parsed from table trimmer section 380. If no data is identified for removal, then the process bypasses step 866 and continues at step 870.

[0065] In step 870, the process parses instructions from presentation section 390 as described above in reference to FIG. 7. In step 875, the process uses this information to inquire whether any of the data currently stored in either the short-term or the long-term database should be rendered and displayed to a user. If data is to be displayed, then in step 876 the process provides the appropriate API for the selected data and causes it to be displayed according to the instructions parsed from presentation section 390. If no data is to be displayed, then the process bypasses step 876 and continues in step 880.

[0066] At step 880, the XML configuration file has been fully executed and the process inquires whether any other XML configuration files are available for execution. If so, the process returns to step 810 and begins again. If not, the process ends.

4.0 IMPLEMENTATION MECHANISMS -- HARDWARE OVERVIEW

[0067] FIG. 9 is a block diagram that illustrates a computer system 900 upon which an embodiment of the invention may be implemented. Computer system 900 includes a bus 902 or other communication mechanism for communicating information, and a processor 904 coupled with bus 902 for processing information. Computer system 900 also includes a main memory 906, such as a random access memory ("RAM") or other dynamic storage device, coupled to bus 902 for storing information and instructions to be executed by processor 904. Main memory 906 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 904. Computer system 900 further includes a read only memory ("ROM") 909 or other static storage device coupled to bus 902 for storing static information and instructions for processor 904. A

storage device 910, such as a magnetic disk or optical disk, is provided and coupled to bus 902 for storing information and instructions.

[0068] Computer system 900 may be coupled via bus 902 to a display 912, such as a cathode ray tube ("CRT"), for displaying information to a computer user. An input device 914, including alphanumeric and other keys, is coupled to bus 902 for communicating information and command selections to processor 904. Another type of user input device is cursor control 916, such as a mouse, trackball, stylus, or cursor direction keys for communicating direction information and command selections to processor 904 and for controlling cursor movement on display 912. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0069] The invention is related to the use of computer system 900 for collecting, aggregating and monitoring network management information. According to one embodiment of the invention, collecting, aggregating and monitoring network management information is provided by computer system 900 in response to processor 904 executing one or more sequences of one or more instructions contained in main memory 906. Such instructions may be read into main memory 906 from another computer-readable medium, such as storage device 910. Execution of the sequences of instructions contained in main memory 906 causes processor 904 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0070] The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 904 for execution. Such a medium may

take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 910. Volatile media includes dynamic memory, such as main memory 906. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 902. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infrared data communications.

[0071] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0072] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 904 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 900 can receive the data on the telephone line and use an infrared transmitter to convert the data to an infrared signal. An infrared detector can receive the data carried in the infrared signal and appropriate circuitry can place the data on bus 902. Bus 902 carries the data to main memory 906, from which processor 904 retrieves and executes the instructions. The instructions received by main memory 906 may optionally be stored on storage device 910 either before or after execution by processor 904.

[0073] Computer system 900 also includes a communication interface 919 coupled to bus 902. Communication interface 919 provides a two-way data communication coupling to a

network link 920 that is connected to a local network 922. For example, communication interface 919 may be an integrated services digital network ("ISDN") card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 919 may be a local area network ("LAN") card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 919 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0074] Network link 920 typically provides data communication through one or more networks to other data devices. For example, network link 920 may provide a connection through local network 922 to a host computer 924 or to data equipment operated by an Internet Service Provider ("ISP") 926. ISP 926 in turn provides data communication services through the worldwide packet data communication network now commonly referred to as the "Internet" 929. Local network 922 and Internet 929 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 920 and through communication interface 919, which carry the digital data to and from computer system 900, are exemplary forms of carrier waves transporting the information.

[0075] Computer system 900 can send messages and receive data, including program code, through the network(s), network link 920 and communication interface 919. In the Internet example, a server 930 might transmit a requested code for an application program through Internet 929, ISP 926, local network 922 and communication interface 919. In accordance with the invention, one such downloaded application provides for collecting, aggregating and monitoring network management information as described herein.

[0076] The received code may be executed by processor 904 as it is received, and/or stored in storage device 910, or other non-volatile storage for later execution. In this manner, computer system 900 may obtain application code in the form of a carrier wave.

5.0 EXTENSIONS AND ALTERNATIVES

[0077] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.
